Hidden Markov Models I

Machine Learning 10-601B

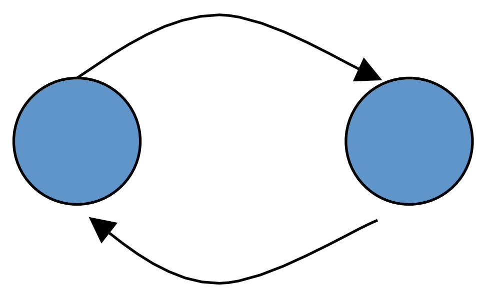
Seyoung Kim

Many of these slides are derived from Tom Mitchell, Ziv Bar-Joseph. Thanks!

What's wrong with Bayesian networks

- Bayesian networks are very useful for modeling joint distributions
- But they have their limitations:
  - Cannot account for temporal / sequence models
  - DAG's (no self or any other loops)

This is not a valid Bayesian network!



Hidden Markov models

- Model a set of observation with a set of hidden states

  - Robot movement

  Observations: range sensor, visual sensor

  Hidden states: location (on a map)

Hidden Markov models

- Model a set of observation with a set of hidden states

  - Robot movement

  Observations: range sensor, visual sensor

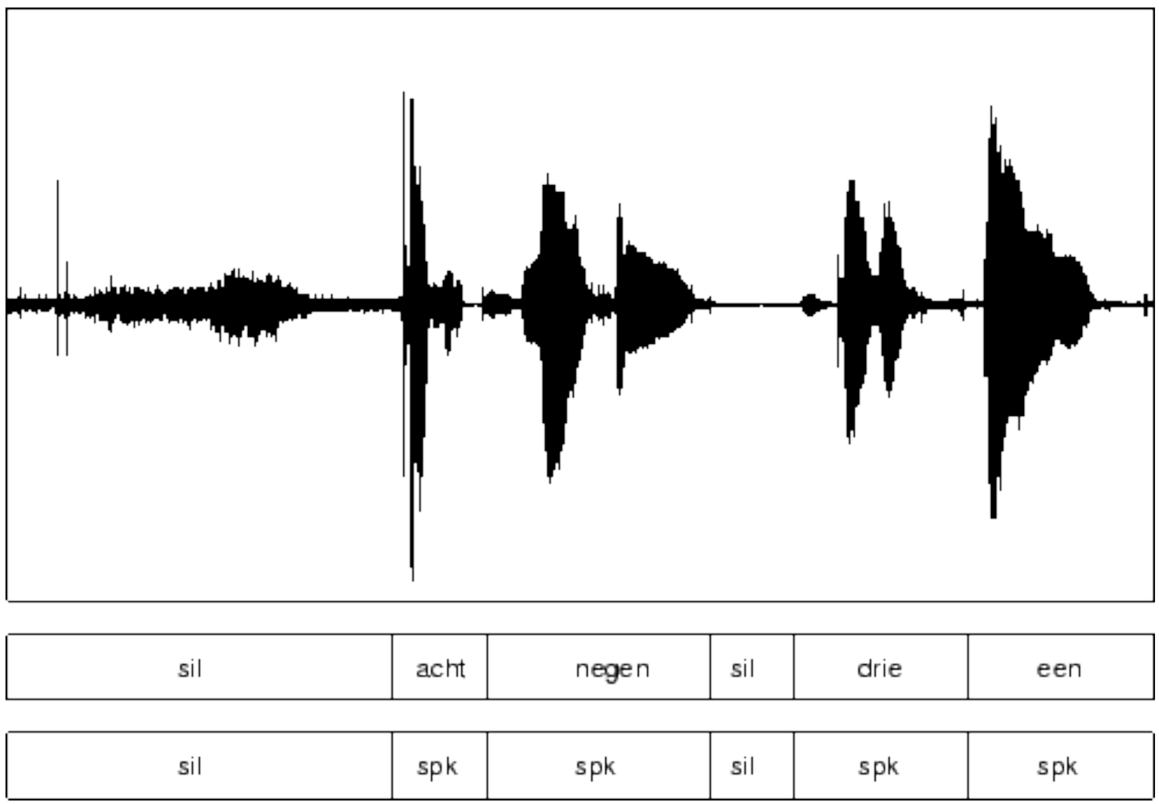
  Hidden states: location (on a map)

1. Hidden states generate observations
2. Hidden states transition to other hidden states

Examples: Speech processing

Speech processing

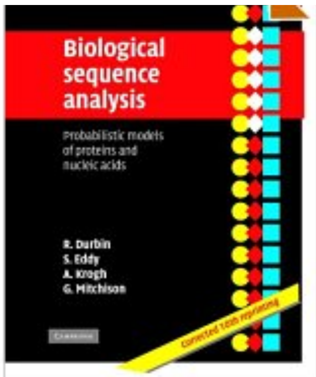
- Observations: sound signals
- Hidden states: parts of speech, words

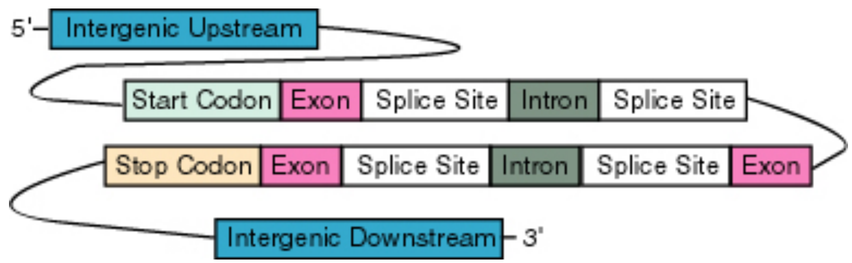


Example: Biological data

Biology

- Observations: DNA base pairs
- Hidden states: Genes



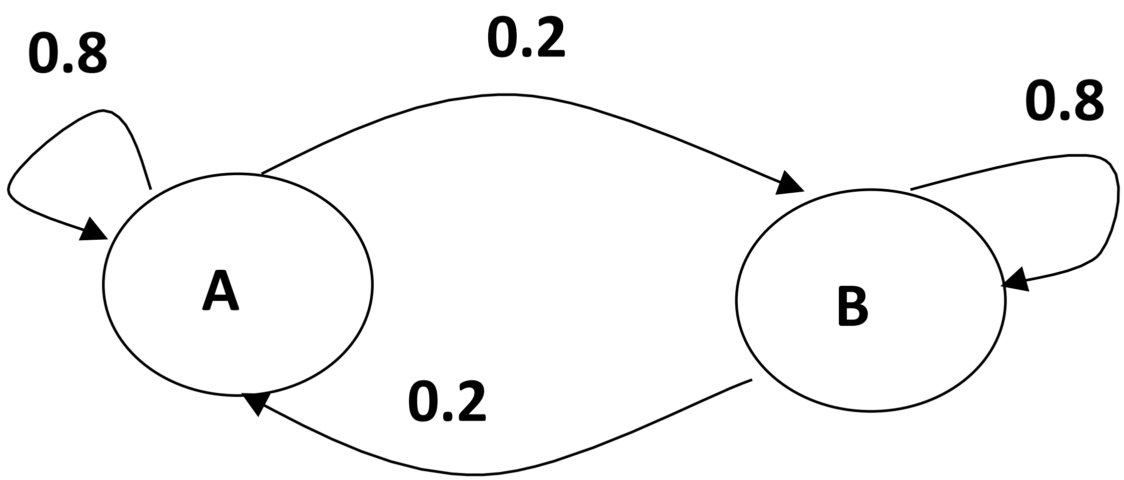


ATGAAGCTACTGTCTTCTATCGAACAAGCATGCGATATTTGCCG
ACTTAAAAAGCTCAAG
TGCTCCAAAGAAAAACCGAAGTGCGCCAAGTGTCTGAAGAA
CAACTGGGAGTGTCGCTAC
TCTCCCAAAACCAAAAGGTCTCCGCTGACTAGGGCACATCTG
ACAGAAGTGGAATCAAGG
CTAGAAAGACTGGAACAGCTATTTCTACTGATTTTTCCTCGAG
AAGACCTTGACATGATT

Example: Gambling on dice outcome

- Two dice A and B, both skewed (output model).
- Can either stay with the same die or switch to the second die (transition model).

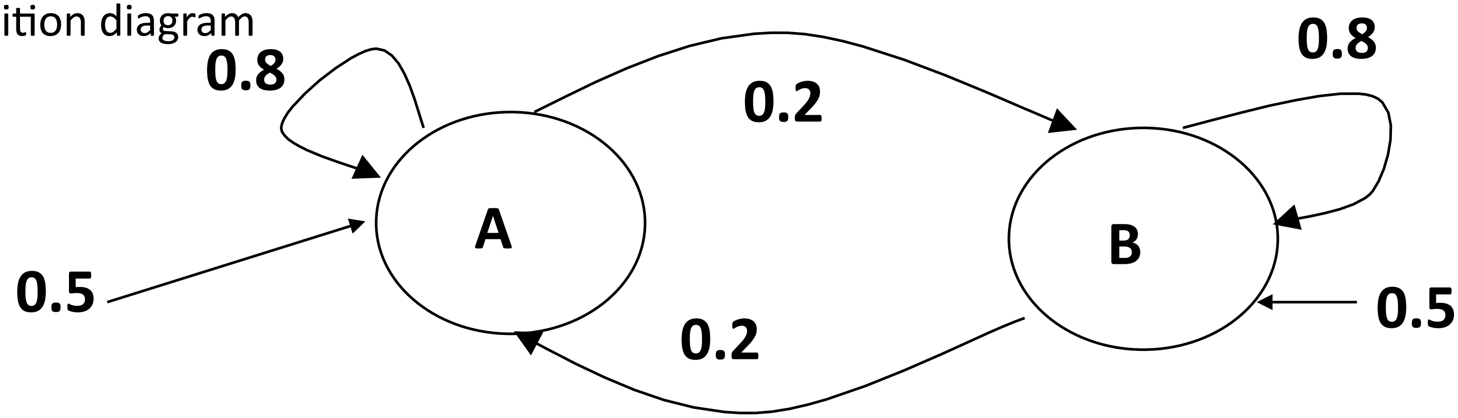
State transition diagram



A Hidden Markov Model

- A set of states $S = \{s_1 \dots s_n\}$
  - In each time point we are in exactly one of these states
- A set of possible outputs $\Sigma=\{\sigma_1, \dots, \sigma_m\}$
  - In each time point we emit a symbol $\sigma_j \in \Sigma$

State transition diagram

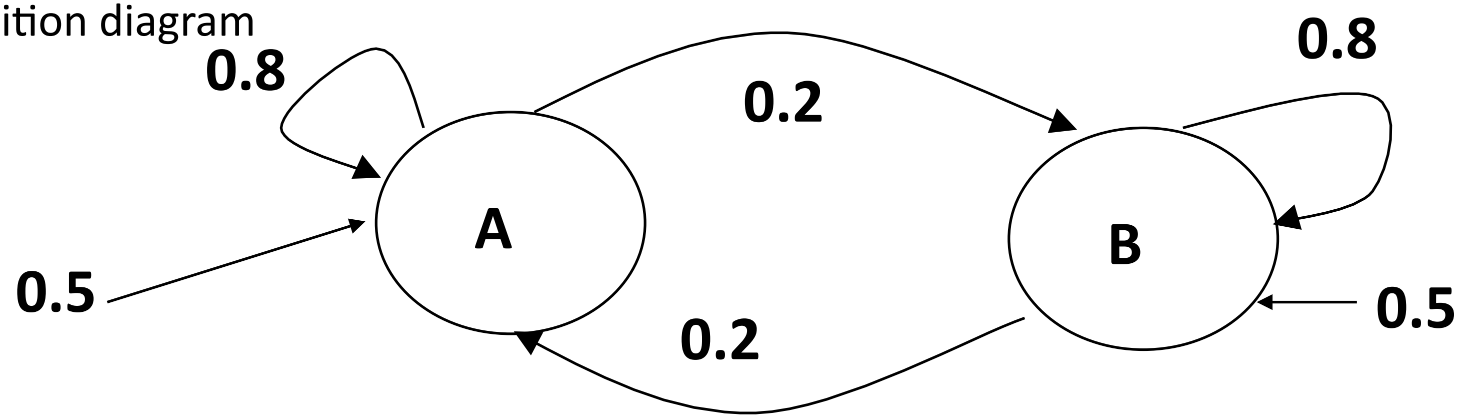


A Hidden Markov Model

- A set of states $S = \{s_1 \dots s_n\}$
  - In each time point we are in exactly one of these states
- A set of possible outputs $\Sigma=\{\sigma_1, \dots, \sigma_m\}$
  - In each time point we emit a symbol $\sigma_j \in \Sigma$

States:	A A A B B B B B B A A
Observations:	1 2 2 1 1 2 1 1 1 2 2

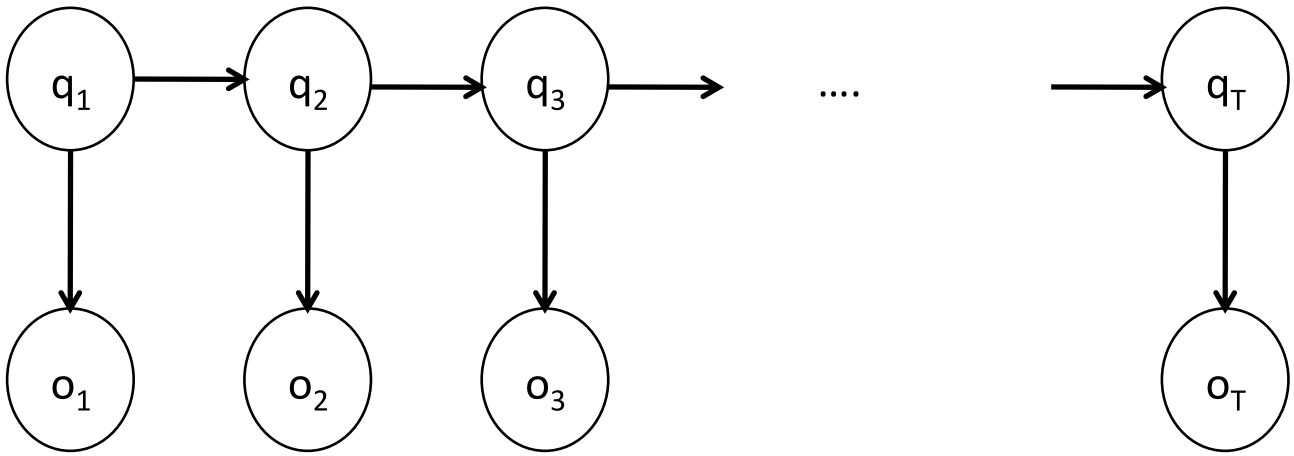
State transition diagram



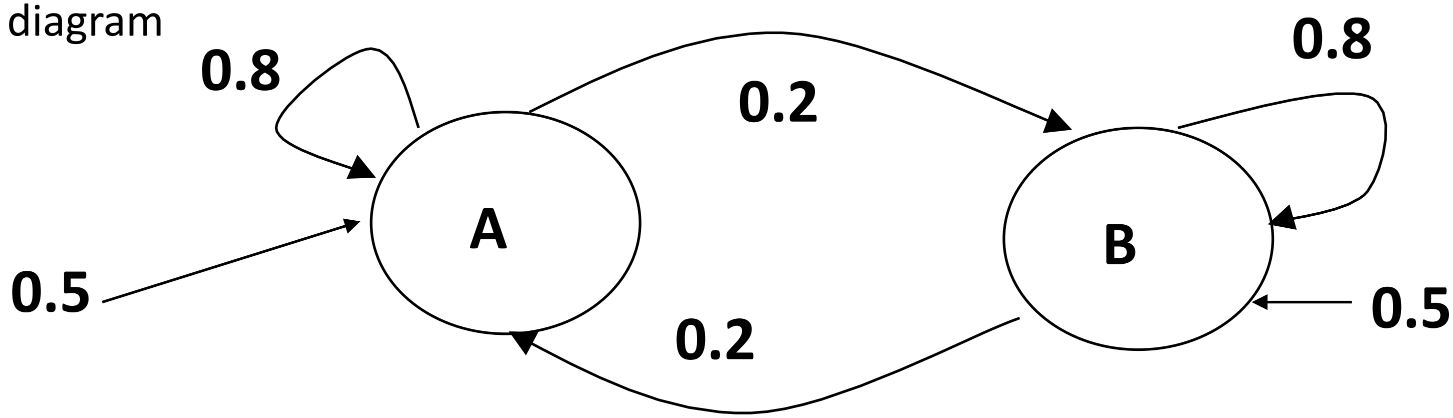
A Hidden Markov Model

- Probabilistic graphical models

States: A A A B B B B B B A A
Observations: 1 2 2 1 1 2 1 1 1 2 2



State transition diagram

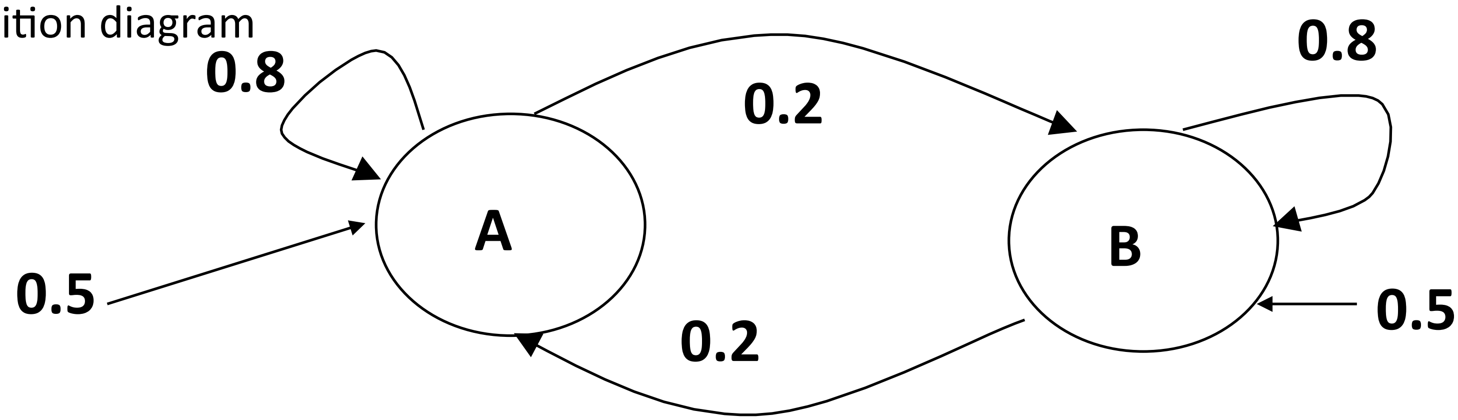


A Hidden Markov Model

- A set of states $S=\{s_1 \dots s_n\}$
  - In each time point we are in exactly one of these states
- A set of possible outputs $\Sigma=\{\sigma_1, \dots, \sigma_m\}$
  - In each time point we emit a symbol $\sigma_j \in \Sigma$

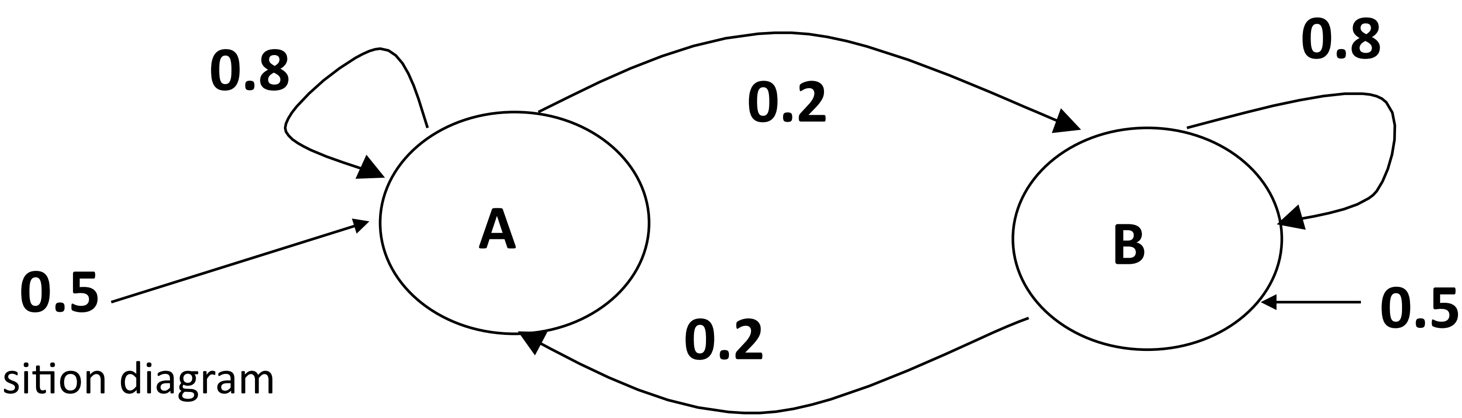
- Random variables
  - States at each time point $Q = \{q_1, \dots, q_T\}$
    - Each q_t can take on values from $\{s_1 \dots s_n\}$
  - Outputs at each time point $O = \{o_1, \dots, o_T\}$
    - Each o_t can take on values from Σ

State transition diagram



A Hidden Markov Model

- Parameters of the model
  - $\Pi_i = \{\pi_1, \ldots, \pi_n\}$: initial state probabilities $P(q_1=s_i)$
    - the probability that we *start* at state $s_{i,}$ i=1,…,n
  - A transition probability model, $P(q_t = s_i \mid q_{t-1} = s_j)$
    - nxn matrix of transition probabilities
  - An emission probability model, $p(o_t = \sigma_j \mid q_t = s_i)$
    - nxm matrix of emission probabilities



State transition diagram

A Hidden Markov Model

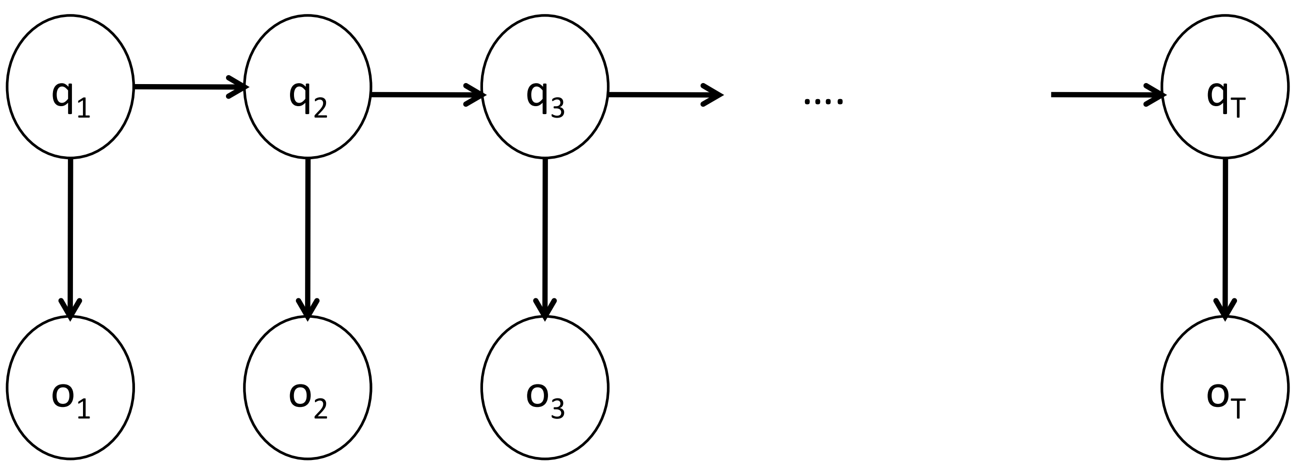
- The joint probability of (Q,O) is defined as

$$P(Q,O) = p(q_1)\prod_{t=1}^{T} p(q_t \mid q_{t-1})\, p(o_t \mid q_t)$$

Initial probability

transition probability

emission probability



A Hidden Markov Model

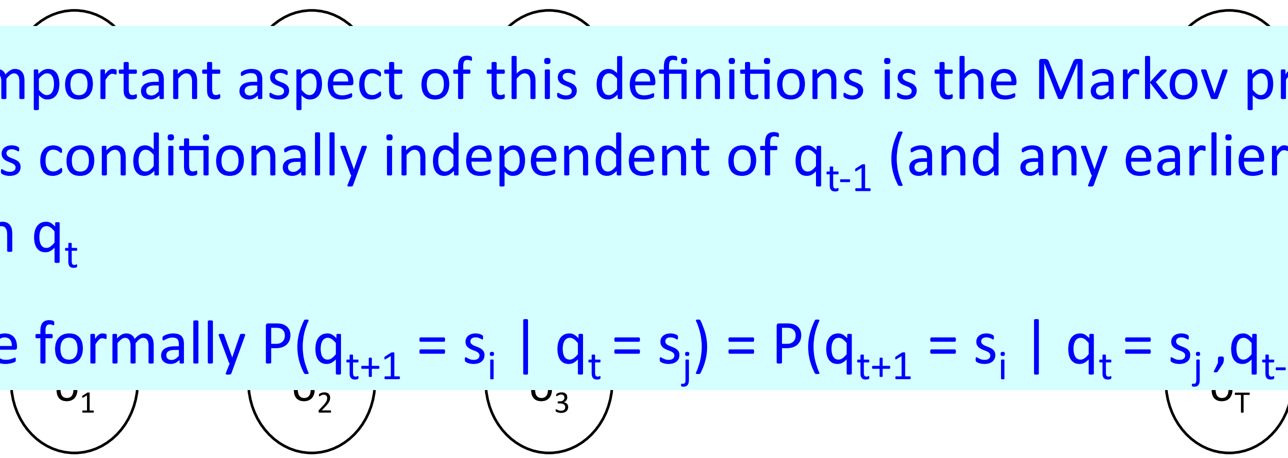
- The joint probability of (Q,O) is defined as

$$P(Q,O) = p(q_1)p(o_1 \mid q_1)\prod_{t=2}^{T} p(q_t \mid q_{t-1})p(o_t \mid q_t)$$

Initial probability

transition probability

emission probability

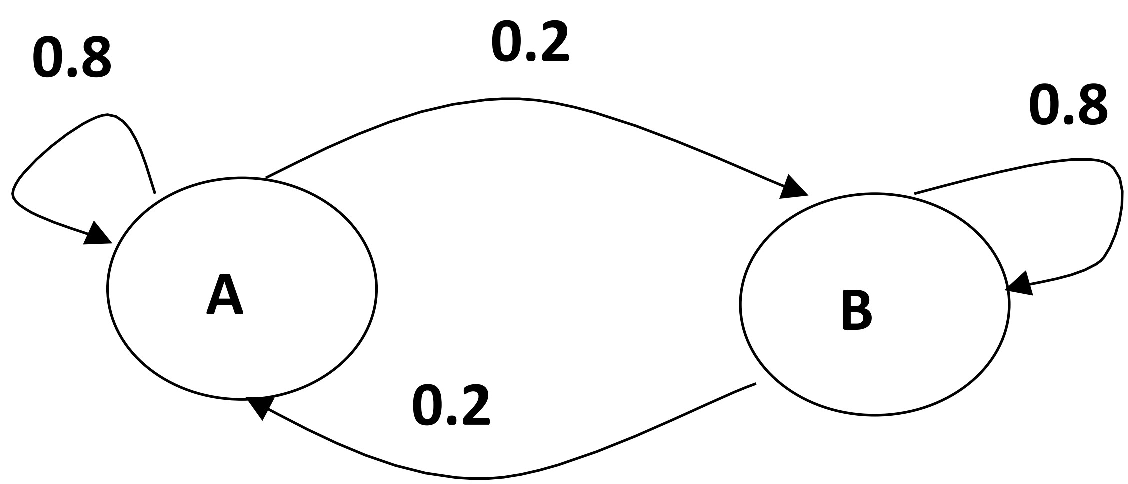


An important aspect of this definitions is the Markov property: q_{t+1} is conditionally independent of q_{t-1} (and any earlier time points) given q_t

More formally $P(q_{t+1} = s_i \mid q_t = s_j) = P(q_{t+1} = s_i \mid q_t = s_j, q_{t-1} = s_j)$

What can we ask when using a HMM?

- A few examples:
  - “Which die is currently being used?”
  - “What is the probability of a 6 in the next role?”
  - “What is the probability of 6 in any of the next 3 roles?”



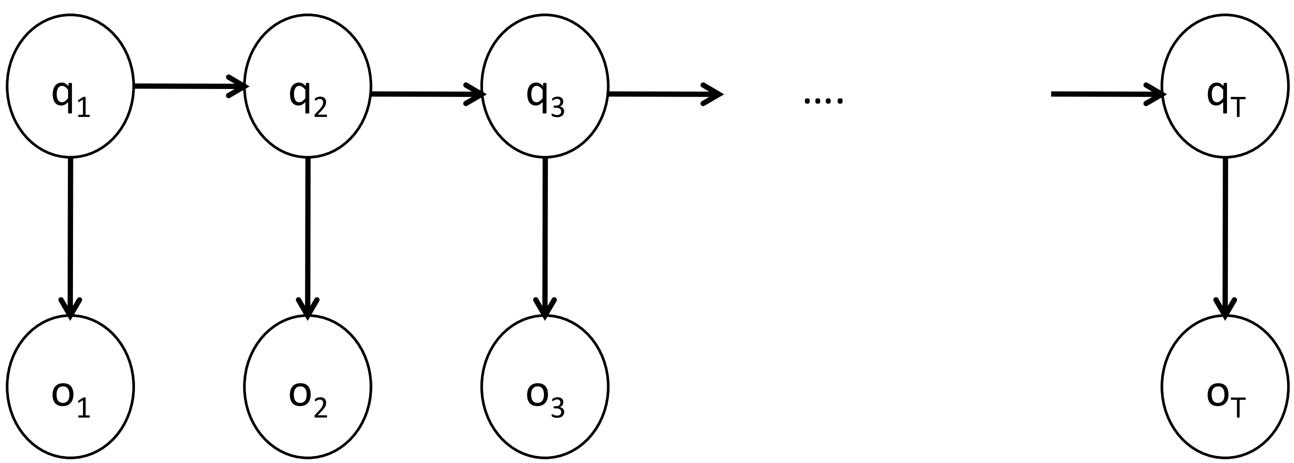
Inference in HMMs

- Computing $P(Q)$ and $P(q_t = s_i)$

  - If we cannot look at observations
- Computing $P(Q \mid O)$ and $P(q_t = s_i \mid O)$

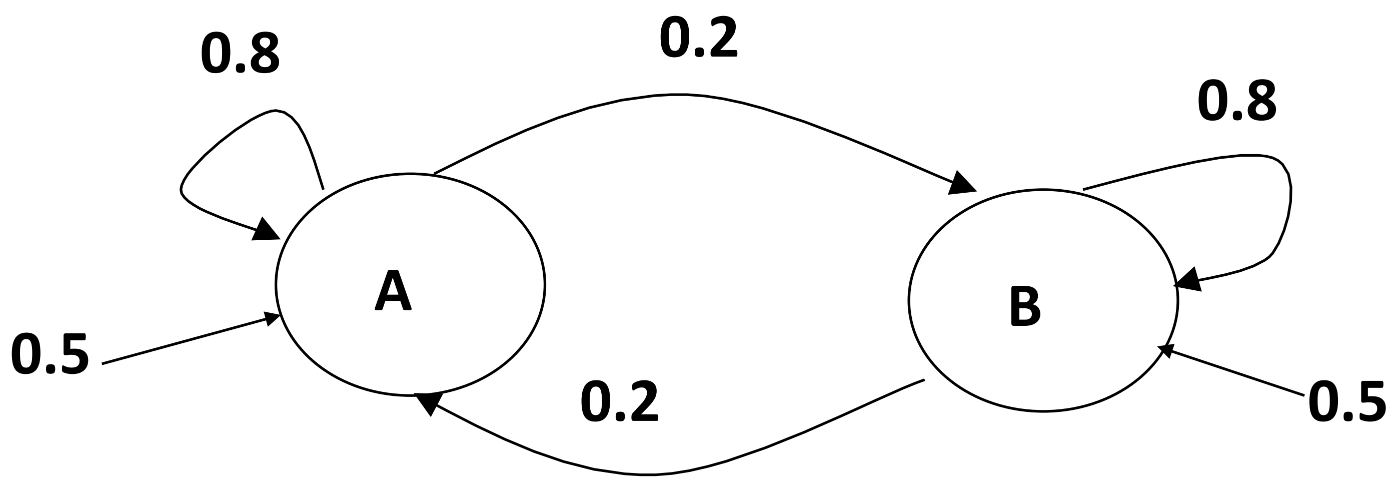
  - When we have observation and care about the last state only
- Computing $\text{argmax}_Q P(Q \mid O)$

  - When we care about the entire path



Which die is currently being used?

- We played *t* rounds so far
- We want to determine $P(q_t = A)$
- Let's assume for now that we cannot observe any outputs (we are blind folded)
- How can we compute this?



$P(q_t = A)$?

- Simple answer: Consider "all" paths that end in A. For each such path Q, let's determine P(Q)

$Q = q_1, \ldots q_{t-1}, A$

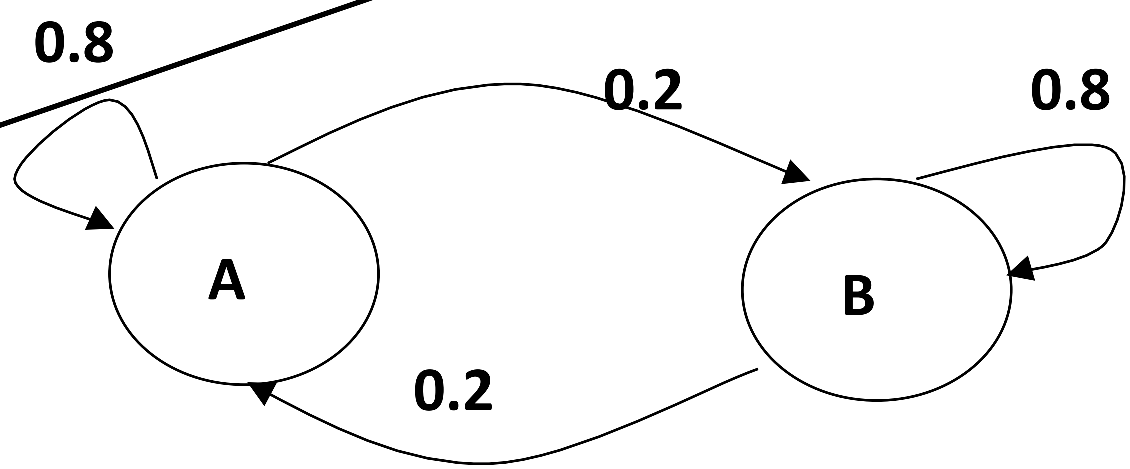
$P(Q) = P(q_1, \ldots q_{t-1}, A) = P(A \mid q_1, \ldots q_{t-1})\, P(q_1, \ldots q_{t-1})$

$= P(A \mid q_{t-1})\, P(q_1, \ldots q_{t-1})$

$= P(A \mid q_{t-1}) \ldots P(q_2 \mid q_1)\, P(q_1)$

Markov property!

Initial probability



$P(q_t = A)$?

- Simple answer:

1. Let's determine P(Q) where Q is any path that ends in A

$Q = q_1, \ldots q_{t-1}, A$

$P(Q) = P(q_1, \ldots q_{t-1}, A)$

$= P(A \mid q_1, \ldots q_{t-1})\, P(q_1, \ldots q_{t-1})$

$= P(A \mid q_{t-1})\, P(q_1, \ldots q_{t-1})$

$= P(A \mid q_{t-1}) \ldots P(q_2 \mid q_1)\, P(q_1)$

2. $P(q_t = A) = \Sigma P(Q)$

where the sum is over all sets of t states
that end in A

$P(q_t = A)$?

- Simple answer:

1. Let's determine P(Q) where Q is any path that ends in A

$Q = q_1, \ldots q_{t-1}, A$

$P(Q) = P(q_1, \ldots q_{t-1}, A)$

$= P(A \mid q_1, \ldots q_{t-1})\, P(q_1, \ldots q_{t-1})$

$= P(A \mid q_{t-1})\, P(q_1, \ldots q_{t-1})$

$= P(A \mid q_{t-1}) \ldots P(q_2 \mid q_1)\, P(q_1)$

2. $P(q_t = A) = \Sigma P(Q)$

where the sum is over all sets of t states that end in A

Q: How many sets Q are there?

A: A lot! (2^{t-1})

Not a feasible solution

$P(q_t = A)$, the smart way

- Let's define $p_t(i)$ as the probability of being in state *i* at time t: $p_t(i) = p(q_t = s_i)$
- We can determine $p_t(i)$ by induction

1. $p_1(i) = \Pi_i$
2. $p_t(i) = ?$

$P(q_t = A)$, the smart way

- Let's define $p_t(i)$ = probability state i at time t = $p(q_t = s_i)$
- We can determine $p_t(i)$ by induction

  1. $p_1(i) = \Pi_i$
  2. $p_t(i) = \Sigma_j \, p(q_t = s_i \mid q_{t-1} = s_j) p_{t-1}(j)$

$P(q_t = A)$, the smart way

- Lets define $p_t(i)$ = probability state i at time t = $p(q_t = s_i)$
- We can determine $p_t(i)$ by induction

  1. $p_1(i) = \Pi_i$
  2. $p_t(i) = \Sigma_j \, p(q_t = s_i \mid q_{t-1} = s_j) p_{t-1}(j)$

This type of computation is called dynamic programming

Complexity: $O(n^2 * t)$

Number of states in our HMM

Time / state	t1	t2	t3
s1	.3		
s2	.7		

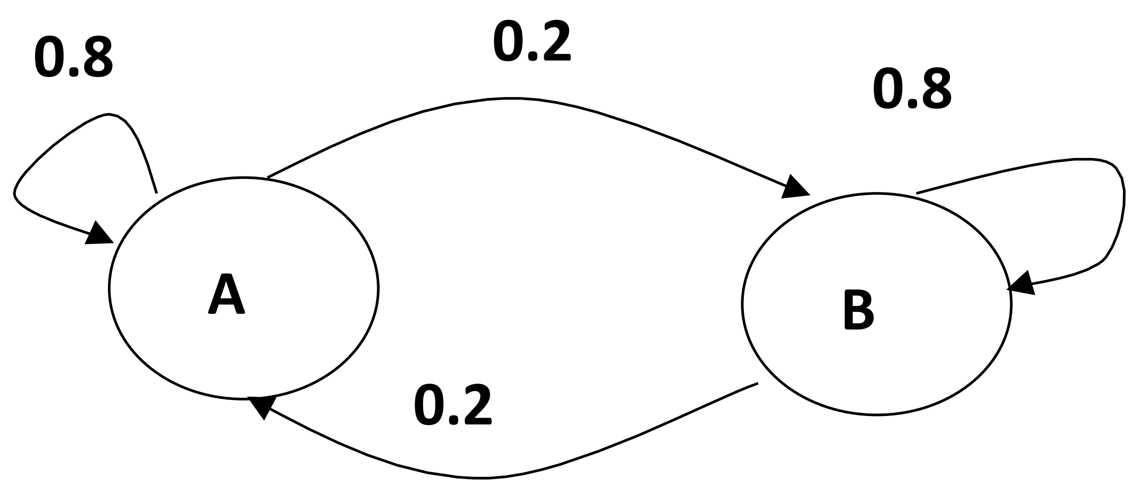
Inference in HMMs

- Computing $P(Q)$ and $P(q_t = s_i)$ √
- Computing $P(Q \mid O)$ and $P(q_t = s_i \mid O)$
- Computing $\text{argmax}_Q P(Q)$

But what if we observe outputs?

- So far, we assumed that we could not observe the outputs
- In reality, we almost always can.

v	P(v |A)	P(v |B)
1	.3	.1
2	.2	.1
3	.2	.1
4	.1	.2
5	.1	.2
6	.1	.3



But what if we observe outputs?

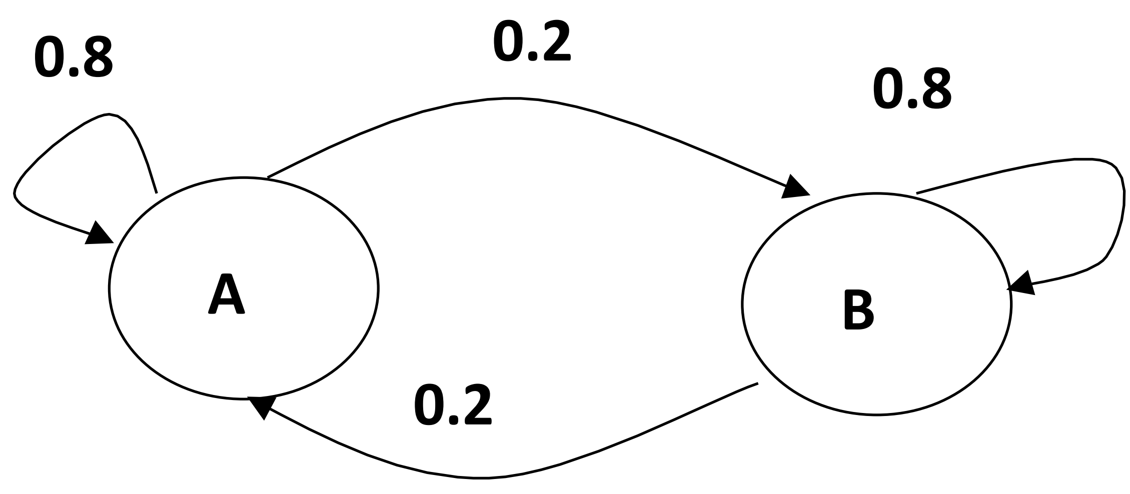
- So far, we assumed that we could not observe the outputs
- In reality, we almost always can.

Does observing the sequence

5, 6, 4, 5, 6, 6

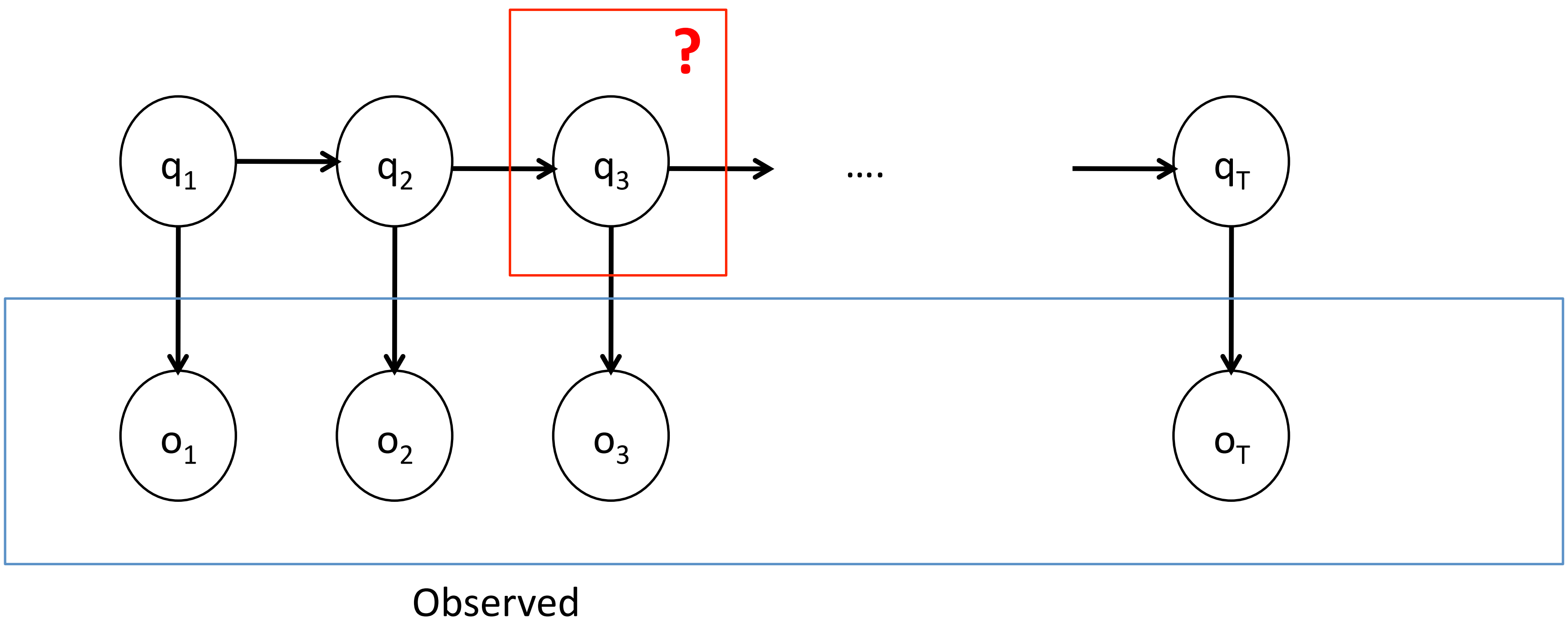
Change our belief about the state?

v	P(v |A)	P(v |B)
1	.3	.1
2	.2	.1
3	.2	.1
4	.1	.2
5	.1	.2
6	.1	.3



When outputs are observed

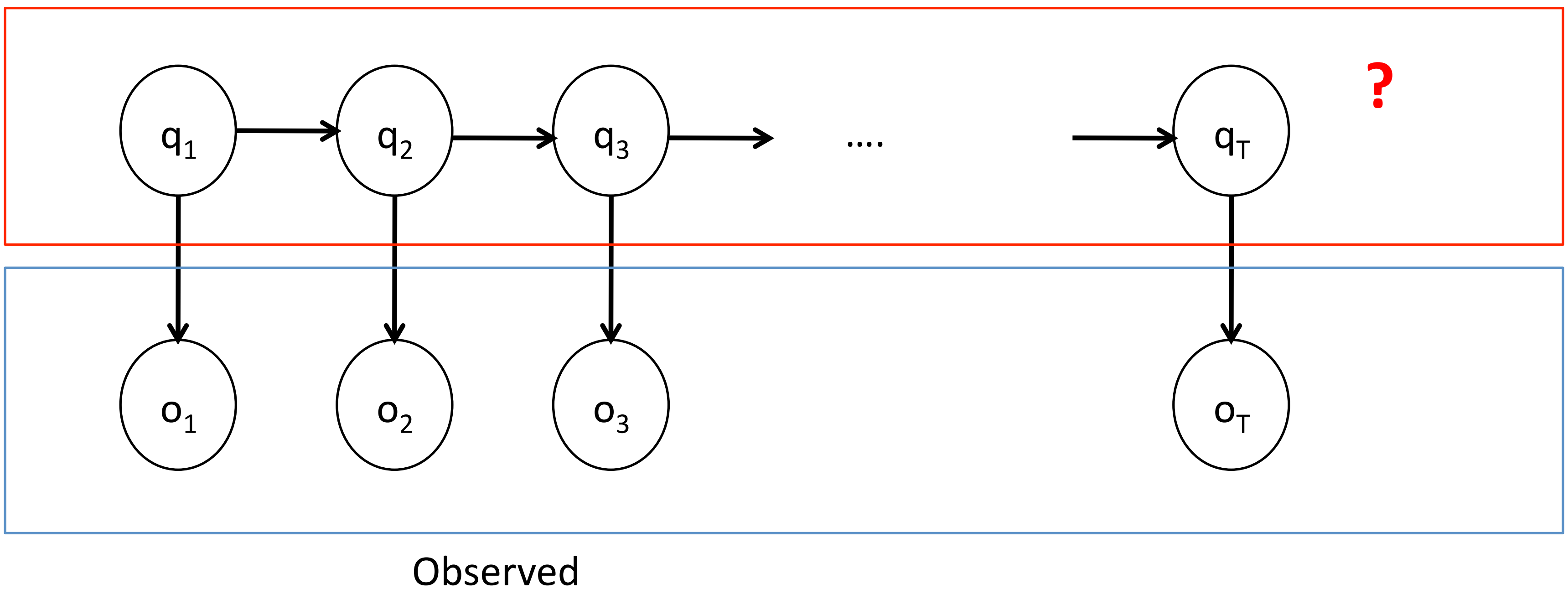
- We want to compute $P(q_t = A \mid O_1 \dots O_t)$



$P(q_t = A|O)$ when outputs are observed

- We want to compute $P(q_t = A \mid O_1 \ldots O_t)$
- Let's start with a simpler question. Given a sequence of states Q, what is $P(Q \mid O_1 \ldots O_t) = P(Q \mid O)$?

- It is pretty simple to move from P(Q) to $P(q_t = A)$



$P(q_t = A|O)$ when outputs are observed

- We want to compute $P(q_t = A \mid O_1 \ldots O_t)$
- Let's start with a simpler question. Given a sequence of states Q, what is $P(Q \mid O_1 \ldots O_t) = P(Q \mid O)$?

  - It is pretty simple to move from P(Q) to $P(q_t = A)$

  - In some cases P(Q) is the more important question

    - Speech processing

    - NLP

P(Q | O)

- We can use Bayes rule:

$$P(Q|O) = \frac{P(O \mid Q)P(Q)}{P(O)}$$

Easy, $P(O \mid Q) = P(o_1 \mid q_1)\ P(o_2 \mid q_2) \ldots P(o_t \mid q_t)$

P(Q | O)

- We can use Bayes rule:

$$P(Q|O) = \frac{P(O|Q)P(Q)}{P(O)}$$

Easy, $P(Q) = P(q_1)\ P(q_2 \mid q_1) \ldots P(q_t \mid q_{t-1})$

P(Q | O)

- We can use Bayes rule:

$$P(Q|O) = \frac{P(O \mid Q)P(Q)}{P(O)}$$

Hard!

P(O)

- What is the probability of seeing a set of observations:
  - An important question in it own rights, for example classification using two HMMs H_1 and H_2
    - Compute $P(O|H_1)$ and $P(O|H_1)$, classify to the model with higher probability

P(O)

- Define $\alpha_t(i) = P(o_1, o_2 \ldots, o_t \wedge \ q_t = s_i)$
- $\alpha_t(i)$ is the probability that we:

    1. Observe $o_1, o_2 \ldots, o_t$
    2. End up at state i

How do we compute α_t (i)?

When outputs are observed

- We want to compute $P(q_t = A \mid O_1 \ldots O_t)$
- For ease of writing we will use the following notations (commonly used in the literature)
  - $a_{j,i} = P(q_t = s_i \mid q_{t-1} = s_j)$
  - $b_i(o_t) = P(o_t \mid s_i)$

Transition probability

Emission probability

Computing $\alpha_t(i)$

- $\alpha_1(i) = P(o_1 \wedge q_1 = i) = P(o_1 \mid q_1 = s_i)\Pi_i$

We must be at a state in time t

chain rule

Markov property

Computing $\alpha_t(i)$

- $\alpha_1(i) = P(o_1 \wedge q_1 = i) = P(o_1 \mid q_1 = s_i)\Pi_I$

We must be at a state in time t

chain rule

Markov property

$$\alpha_{t+1}(i) = P(O_1 \ldots O_{t+1} \wedge q_{t+1} = s_i) =$$

$$\sum_j P(O_1 \ldots O_t \wedge q_t = s_j \wedge O_{t+1} \wedge q_{t+1} = s_i) =$$

$$\sum_j P(O_{t+1} \wedge q_{t+1} = s_i \mid O_1 \ldots O_t \wedge q_t = s_j) P(O_1 \ldots O_t \wedge q_t = s_j) =$$

$$\sum_j P(O_{t+1} \wedge q_{t+1} = s_i \mid O_1 \ldots O_t \wedge q_t = s_j) \alpha_t(j) =$$

$$\sum_j P(O_{t+1} \mid q_{t+1} = s_i) P(q_{t+1} = s_i \mid q_t = s_j) \alpha_t(j) =$$

$$\sum_j b_i(O_{t+1}) a_{j.i} \alpha_t(j)$$

Example: Computing $\alpha_3(B)$

- We observed 2,3,6

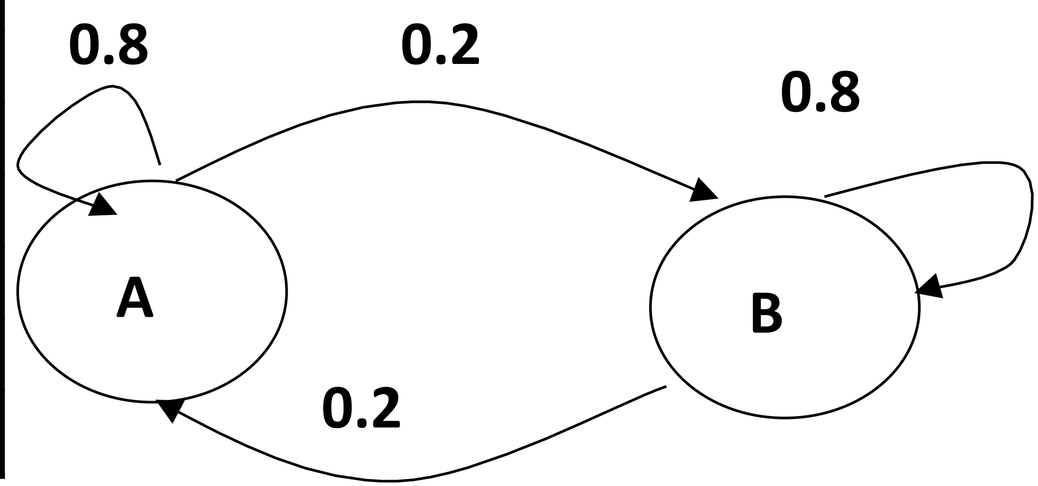
$\alpha_1(A) = P(2 \wedge q_1 = A) = P(2 \mid q_1 = A)\Pi_A = .2*.7 = .14,\ \alpha_1(B) = .1*.3 = .03$

$\alpha_2(A) = \Sigma_{j=A,,B} b_A(3) a_{j,A}\, \alpha_1(j) = .2*.8*.14+.2*.2*.03 = 0.0236,\ \alpha_2(B) = 0.0052$

$\alpha_3(B) = \Sigma_{j=A,,B} b_B(6) a_{j,B}\, \alpha_2(j) = .3*.2*.0236+.3*.8*.0052 = 0.00264$

$\Pi_A=0.7$
$\Pi_b=0.3$

v	P(v |A)	P(v |B)
1	.3	.1
2	.2	.1
3	.2	.1
4	.1	.2
5	.1	.2
6	.1	.3



Where we are

- We want to compute P(Q | O)
- For this, we only need to compute P(O)
- We know how to compute $\alpha_t(i)$

From now its easy

$\alpha_t(i) = P(o_1, o_2 \ldots, o_t \wedge q_t = s_i)$

so

$P(O) = P(o_1, o_2 \ldots, o_t) = \Sigma_i P(o_1, o_2 \ldots, o_t \wedge q_t = s_i) = \Sigma_i \alpha_t(i)$

note that

$$p(q_t = s_i \mid o_1, o_2 \ldots, o_t) = \frac{\alpha_t(i)}{\sum_j \alpha_t(j)}$$

P(A | B) = P(A ∧ B) / P(B)

Complexity

- How long does it take to compute P(Q | O)?
  - P(Q): O(n)
  - P(O|Q): O(n)
  - P(O): $O(n^2t)$

Inference in HMMs

- Computing $P(Q)$ and $P(q_t = s_i)$ √
- Computing $P(Q \mid O)$ and $P(q_t = s_i \mid O)$ √
- Computing $\text{argmax}_Q P(Q)$

Most probable path

- We are almost done …
- One final question remains

How do we find the most probable path, that is Q* such that

$$P(Q^* \mid O) = \text{argmax}_Q P(Q|O)?$$

- This is an important path
  - The words in speech processing
  - The set of genes in the genome etc.

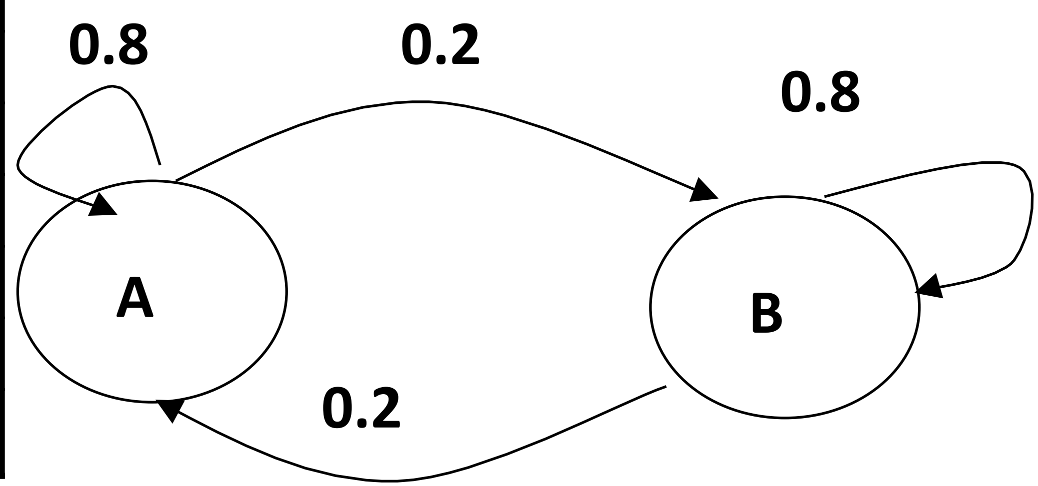
Example

- What is the most probable set of states leading to the sequence:

  1,2,2,5,6,5,1,2,3 ?

Π_A=0.7
Π_b=0.3

v	P(v |A)	P(v |B)
1	.3	.1
2	.2	.1
3	.2	.1
4	.1	.2
5	.1	.2
6	.1	.3



Most probable path

$$\arg\max_Q P(Q \mid O) = \arg\max_Q \frac{P(O \mid Q)P(Q)}{P(O)}$$

$$= \arg\max_Q P(O \mid Q)P(Q)$$

We will use the following definition:

$$\delta_t(i) = \max_{q_1 \ldots q_{t-1}} p(q_1 \ldots q_{t-1} \wedge q_t = s_i \wedge O_1 \ldots O_t)$$

In other words we are interested in the most likely path from 1 to t that:

1. Ends in S_i
2. Produces outputs $O_1 \ldots O_t$

Computing $\delta_t(i)$

$$\delta_t(i) = \max_{q_1 \ldots q_{t-1}} p(q_1 \ldots q_{t-1} \wedge q_t = s_i \wedge O_1 \ldots O_t)$$

Initialization at t=1

$$\begin{aligned}\delta_1(i) &= p(q_1 = s_i \wedge O_1) \\ &= p(q_1 = s_i) p(O_1 \mid q_1 = s_i) \\ &= \pi_i b_i(O_1)\end{aligned}$$

Q: Given $\delta_t(i)$, how can we compute $\delta_{t+1}(i)$?

A: To get from $\delta_t(i)$ to $\delta_{t+1}(i)$ we need to

1. Add an emission for time t+1 (O_{t+1})
2. Transition to state s_i

$$\begin{aligned}\delta_{t+1}(i) &= \max_{q_1 \ldots q_t} p(q_1 \ldots q_t \wedge q_{t+1} = s_i \wedge O_1 \ldots O_{t+1}) \\ &= \max_j \delta_t(j) p(q_{t+1} = s_i \mid q_t = s_j) p(O_{t+1} \mid q_{t+1} = s_i) \\ &= \max_j \delta_t(j) a_{j,i} b_i(O_{t+1})\end{aligned}$$

The Viterbi algorithm

$$\delta_{t+1}(i) = \max_{q_1 \ldots q_t} p(q_1 \ldots q_t \wedge q_{t+1} = s_i \wedge O_1 \ldots O_{t+1})$$

$$= \max_j \delta_t(j) p(q_{t+1} = s_i \mid q_t = s_j) p(O_{t+1} \mid q_{t+1} = s_i)$$

$$= \max_j \delta_t(j) a_{j,i} b_i(O_{t+1})$$

- Once again we use dynamic programming for solving $\delta_t(i)$
- Once we have $\delta_t(i)$, we can solve for our $P(Q^*|O)$ by:

$P(Q^* \mid O) = \text{argmax}_Q P(Q|O) =$

path defined by $\text{argmax}_j \ \delta_t(j)$,

Inference in HMMs

- Computing $P(Q)$ and $P(q_t = s_i)$ √
- Computing $P(Q \mid O)$ and $P(q_t = s_i \mid O)$ √
- Computing $\text{argmax}_Q P(Q)$ √

What you should know

- Why HMMs? Which applications are suitable?
- Inference in HMMs

  - No observations

  - Probability of next state w. observations

  - Maximum scoring path (Viterbi)